Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Maestría en Telecomunicaciones

Evaluación de un modelo matemático para la elección de métricas en el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s, utilizando NS-3 y RStudio.

Trabajo de Investigación previa a la obtención del título de Magíster en Telecomunicaciones

AUTOR:

Ing. Marilin Salomé Quishpi Flores

DIRECTOR:

Ing. Juan Ochoa Aldeán, Mg. Sc.

Loja – Ecuador

2025

Educamos para Transformar

Certificación

Loja, 03 de abril de 2025

Ing. Juan Ochoa Aldean Mg. Sc

DIRECTOR DE TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del trabajo de Titulación denominado: **Evaluación de un modelo matemático para la elección de métricas en el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s, utilizando NS-3 y RStudio,** previo a la obtención del título de **Magíster en Telecomunicaciones**, de la autoría de la estudiante **Marilin Salome Quishpi Flores** con **cédula de identidad N° 0301549143**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Firmado electrónicamente por:
JUAN GABRIEL OCHOA ALDEAN

Ing. Juan Ochoa Aldean Mg. Sc.

DIRECTOR DE TRABAJO DE INVESTIGACIÓN

Autoría

Yo, **Marilin Salomé Quishpi Flores**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual

Firma: Firmado electrónicamente por: MARILIN SALOME QUISHPI FLORES

Cédula de Identidad: 0301549143

Fecha: 04 de abril de 2025

Correo electrónico: marilin.quishpi@unl.edu.ec

Teléfono o celular: 0984063411

Carta de autorización

Yo, **Marilin Salomé Quishpi Flores**, declaro ser autor del Trabajo de Titulación denominado: **Evaluación de un modelo matemático para la elección de métricas en el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s, utilizando NS-3 y RStudio**, como requisito para optar el título de **Magíster en Telecomunicaciones**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 03 días del mes de abril de dos mil veinticinco.

Firma: Firmado electrónicamente por: MARILIN SALOME QUISHPI FLORES

Autor: Marilin Salomé Quishpi Flores

Cédula: 0301549143

Dirección: Medardo Ángel Silva y Camilo Egas

Correo Electrónico: marilin.quishpi@unl.edu.ec

Celular: 0984063411

DATOS COMPLEMENTARIOS:

Director de trabajo de investigación: Ing. Juan Ochoa Aldean Mg. Sc.

Dedicatoria

A mis amados hijos, **Anthony, Angeline e Isis,** pilares fundamentales de mi vida y razón de mi lucha.

Cada día, su amor se convierte en la energía que alimenta mi perseverancia, su sonrisa es mi motivación para crecer, y su existencia en el más poderoso motivo para seguir aprendiendo.

Esta tesis es un testimonio de que, con objetivos claros y determinación, los sueños se transforman en logros.

Que ella les recuerde siempre que las metas –por grandes que parezcan– se alcanzan paso a paso, con paciencia y convicción.

Porque ustedes son mi mayor inspiración, y esta victoria es también la suya.

Con todo mi amor,

Marilin Salome Quishpi Flores

Agradecimiento

Quiero expresar mi más sincero agradecimiento a todas las personas que me brindaron su apoyo y acompañaron en esta bonita travesía académica. Sin su ayuda, no habría sido posible alcanzar esta meta que hoy con alegría culmino.

A mis hijos, por su comprensión, tolerancia y estímulo constante a lo largo de mis estudios. Su paciencia y cariño fueron mi motor en los momentos más desafiantes.

A mi esposo, **Francisco**, cuyo amor incondicional y apoyo inagotable me sostuvieron en cada etapa. Gracias por ser mi pilar y por creer en mí incluso cuando yo dudaba.

Al **Ing. Juan Ochoa Aldean, Mg. Sc.,** director de esta tesis, por su invaluable guía, dedicación y sabios consejos durante la elaboración de este trabajo. Su experiencia y compromiso fueron faros que iluminaron mi camino académico.

Y, por supuesto, a todas aquellas personas que, de una u otra forma, contribuyeron a la realización de este proyecto. Cada palabra de aliento, cada gesto de ayuda y cada enseñanza recibida quedaron grabados en este logro que hoy comparto con ustedes."

Marilin Salome Quishpi Flores

Índice de Contenidos

Certificación ... ii

Autoría ... iii

Carta de autorización ... iv

Dedicatoria ... v

Agradecimiento ... vi

Índice de Contenidos ... vii

Índice de Tablas ... ix

Índice de Figuras ... x

1. Título ... 1

2. Resumen ... 2

2.1. Abstract ... 2

2.2. Palabras clave ... 2

3. Introducción ... 3

3.1. Objetivos ... 3

3.1.1. Objetivo general ... 3

3.1.2. Objetivos específicos ... 4

4. Marco Teórico ... 4

4.1. Wireless Mesh Networks ... 4

4.1.1. Arquitectura WMN ... 5

4.1.2. Protocolos para WMN ... 7

4.2. Estándar 802.11s (Jerome, 2011) ... 9

4.2.1. HWMP ... 9

4.2.2. Métrica ALM (Airtime Link Metric) ... 10

4.3. Simulador NS-3 ... 11

4.3.1. 802.11s en NS-3 ... 13

4.4. Modelos lineales. ... 14

4.4.1. El modelo lineal ... 15

4.4.2. Análisis del modelo lineal ... 17

4.4.3. Estimación de los parámetros ... 17

4.4.4. La recta de los mínimos cuadrados ... 18

4.4.5. Contribución y apalancamiento 19
4.5. RStudio 19
4.5.1. Conceptos fundamentales 19
4.5.2. El modelo lineal en RStudio 19
4.5.3. Ajuste de modelo lineales con lm () 20
5. Metodología 22
5.1. Descripción de los scripts en NS-3 22
5.1.1. Mesh-ping 22
5.1.2. Airtime-Metric. 26
6. Resultados. 27
6.1. Descripción de Script en RStudio 27
6.1.1. Modelo 1. Modelo de Regresión lineal 31
6.1.2. Modelo 2 Modelo de regresión lineal múltiple con selección de variables . 32
6.1.3. Modelo 3 Modelo propuesto 32
7. Discusión 36
8. Conclusiones 38
9. Recomendaciones 38
1. Referencias bibliográficas. 40
2. Anexos 42

Índice de Tablas

Tabla 1. Métrica AML 11

Tabla 2. Configuración de parámetros de enrutamiento. 25

Tabla 3. Librerías utilizadas en RStudio. 28

Tabla 4. Resultados configuración planteada en NS-3 34

Tabla 5. Resultados configuración modelo propuesto 35

Tabla 6. Resultados comparación entre modelos 36

Tabla 7. Métricas utilizadas para el protocolo HWMP 38

Índice de Figuras

Figura 1. Infraestructura mesh 6
Figura 2. Cliente mesh 6
Figura 3. Mesh híbrido 7
Figura 4. Red mallada nxm 13
Figura 5. Regresión lineal simple. 15
Figura 6. Regresión lineal múltiple. 16
Figura 7. Regresión lineal multivariable 16
Figura 8. Regresión lineal simple 17
Figura 9. Estimación de parámetros 18
Figura 10. Mesh-ping 22
Figura 11. Declaración de librerías. 23
Figura 12. Declaración de variables. 23
Figura 13. Inicialización de variables 24
Figura 14. Posición de nodos. 25
Figura 15. Activación del reporte. 25
Figura 16. Declaración de librerías Airtime-Metric. 26
Figura 17. Parametrización Objeto AirtimeLinkMetric 26
Figura 18. Cálculo 27
Figura 19. Carga de librerías 28
Figura 20. Carga de datos 29
Figura 21. Reducción de Muestras 29
Figura 22. Ejecución del programa 30
Figura 23. Conversión a factores 31
Figura 24. Modelo de regresión lineal 31

Figura 25. Modelo de regresión múltiple 32
Figura 26. Modelo propuesto 32
Figura 27. Validación de modelos de regresión 33
Figura 28. Modelo IEEE 802.11 34
Figura 29. Modelo Propuesto 35

1. Título

Evaluación de un modelo matemático para la elección de métricas en el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s, utilizando NS-3 y RStudio.

2. Resumen

La necesidad de brindar a los usuarios redes de fácil despliegue, a un bajo costo y que les permitan acceder a las altas tasas de transmisión que exigen las aplicaciones multimedia actuales y los servicios en la nube, hace que sea de gran importancia estudiar métodos que permitan mejorar el desempeño del protocolo IEEE 802.11s en diferentes escenarios, haciendo uso de diferentes métricas en el protocolo HWMP.

Tomamos en cuenta que la demanda en los requerimientos de ancho de banda en usuarios móviles tiene un crecimiento exponencial, es importante mejorar el rendimiento de soluciones estandarizadas como la IEEE 802.11s y para ello se puede aplicar modelos matemáticos en la solución de este tipo de problemas, mediante los cuales se pueda realizar el escogitamiento de la métrica o las métricas que permitan optimizar el establecimiento de rutas en diferentes escenarios de aplicación.

2.1. Abstract

The need to provide users with easy-to-deploy, low-cost networks that allow them to access the high transmission rates required by today's multimedia applications and cloud services makes it essential to study methods that improve the performance of the IEEE 802.11s protocol in different scenarios, utilizing different metrics in the HWMP protocol.

Considering that the demand for bandwidth requirements among mobile users is growing exponentially, it is essential to improve the performance of standardized solutions such as IEEE 802.11s. To this end, mathematical models can be applied to solve these problems, enabling the selection of metrics that optimize route establishment in different application scenarios.

2.2. Palabras clave

WMN; 802.11s; HWMP; ALM; NS3; RStudio; LM; GLM.

3. Introducción

Las redes malladas inalámbricas (Wireless Mesh Networks) han tenido un gran éxito y aplicación en el dominio industrial, militar y comercial. Son un dominio rápidamente creciente y esto trae el desafío del enrutamiento efectivo debido a la volatilidad típica de tráfico en topologías complejas.

Muchos estudios han intentado resolver el problema de enrutamiento mediante métodos heurísticos, asumiendo que el tráfico de demandas de la red es estático y conocido de antemano. Como resultado, estos algoritmos tienden a sufrir un desempeño pobre. De hecho, trabajos recientes han demostrado que el tráfico inalámbrico es muy variable y difícil de caracterizar.

Las redes Mesh abiertas son redes ad-hoc descentralizadas que no se basan en infraestructuras previas, como routers o puntos de acceso. En su lugar, cada nodo participa en el enrutado, siendo él mismo un router y enviando datos de otros, y de ese modo la determinación de las rutas se hace dinámicamente.

Es de suma importancia el análisis del rendimiento de diferentes protocolos de comunicación que deben interactuar con diversos dispositivos que hacen al enlace de los nodos de la red a fin de establecer la integración tecnológica disponible. En general, la optimización se basa en lograr el mejor camino para enrutar los paquetes de datos, sin demoras o con una demora mínima en función de lograr un mejor aprovechamiento de los recursos utilizados.

3.1. Objetivos

3.1.1. Objetivo general

Contribuir a mejorar el desempeño del protocolo HWMP, el cual es parte del estándar IEEE 802.11s, a través de la evaluación de modelos matemáticos para la selección de las métricas con la finalidad de optimizar las rutas dentro de una red WMN.

3.1.2. Objetivos específicos

Realizar pruebas de simulación del protocolo HWMP en el simulador NS-3, con la finalidad de obtener variables de desempeño de la red WMN.

Estudiar la implementación de modelos de regresión en RStudio y evaluar su utilización en la implementación de un modelo matemático que permita mejorar el desempeño de HWMP.

Evaluar un modelo matemático que permita decidir la métrica o métricas a utilizar por parte de HWMP para el establecimiento de las rutas, a través de pruebas estadísticas en RStudio.

4. Marco Teórico

4.1. Wireless Mesh Networks

Una Red Mallada Inalámbrica (Mesh) es una red compuesta por nodos organizados en una topología de malla. Son redes en las cuales la información es pasada entre nodos en una forma de todos contra todos y en una jerarquía plana, en contraste a las redes centralizadas.

Las ventajas que presenta frente a otras redes son el bajo costo al utilizar enlaces inalámbricos, la facilidad de aumentar el área de cobertura incluyendo nuevos nodos, la robustez que presenta ante fallos al disponer de rutas alternativas y la capacidad de transmisión que permiten aplicaciones a los usuarios en tiempo real de voz, video y datos. Por tanto, se puede incluir un nuevo nodo en cualquier momento y lugar. Como consecuencia el costo de este tipo de redes inalámbricas es mucho menor que en las redes cableadas, ya que no hay que invertir en materiales de cableado y en estudios enfocados a la unión más óptima de los nodos.

En una red mallada un conjunto de nodos se comunican entre sí de manera directa transmitiendo la información de nodo a nodo hasta que llega a su destino final. La información atraviesa múltiples saltos y no hay necesidad de una unidad centralizada que controle el modo de transmisión. La comunicación se realiza entre los nodos directamente. Cada nodo puede ser origen y destino de los datos o encaminar la información de otros nodos. Las redes malladas inalámbricas son robustas al tener varios caminos disponibles entre el nodo origen

y el destino, de modo que el servicio no se ve afectado por la caída de un nodo o por la ruptura de un enlace.

Dado que la forma de operar que tienen estas redes consiste en que los datos pasan de un nodo a otro hasta que llegan a su destino, los algoritmos de ruteo dinámico necesitan que cada nodo comunique información de ruteo a otros nodos en la red.

Cada nodo determina qué hacer con los datos que recibe, ya sea pasarlos al próximo nodo o quedárselos, dependiendo del protocolo utilizado. El algoritmo de ruteo usado siempre debería asegurar que la información tome el camino más apropiado de acuerdo a una métrica. Una métrica es el valor por el cual los protocolos determinan cuál ruta tomar o con cuál nodo comunicarse.

Una de las debilidades y limitaciones de las redes Mesh es la latencia (el retardo de propagación de los paquetes), que crece con el número de saltos. Los efectos del retardo son dependientes de la aplicación.

Otra debilidad es la disminución del rendimiento en todas las redes multisalto, esto es, a mayor número de saltos, se tiene menor rendimiento.

4.1.1. Arquitectura WMN

La arquitectura de las redes mesh se puede clasificar en tres tipos:

Infraestructura Mesh: La infraestructura está formada por los routers mesh y es el esqueleto de la red. Dichos routers realizarán las funciones de gateway, routing, etc., y permitirán la conexión a Internet. Del mismo modo interconectan todo tipo de redes inalámbricas existentes, como puede ser Wifi, WiMax, telefonía móvil, tal como muestra la Figura 1.

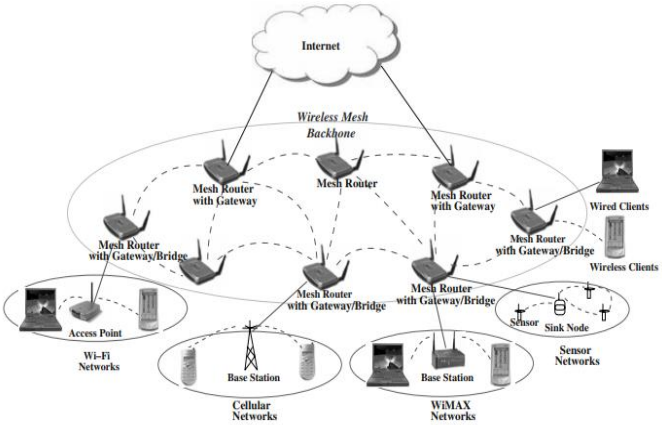

Figura 1. Infraestructura mesh

Aquellos dispositivos que tengan tecnología Ethernet se conectarán a los routers mediante la misma. Para aquellos dispositivos que utilicen la misma tecnología radio que dispongan los routers, se conectarán directamente a ellos, y si es distinta podrán hacerlo mediante sus estaciones base que a su vez utilizarán Ethernet.

Clientes mesh: Los clientes mesh proporcionan una conexión punto a punto entre los dispositivos además de realizar funciones básicas de red, como encaminamiento o configuración. De este modo no es necesario un router mesh. Estos clientes forman una red y sería similar a la conocida ad-hoc. Sin embargo, los clientes mesh disponen de una tecnología superior a los clientes habituales puesto que su software y hardware han de ser capaces de soportar las funciones necesarias para la conexión.

En la Figura 2 se puede apreciar la arquitectura descrita.

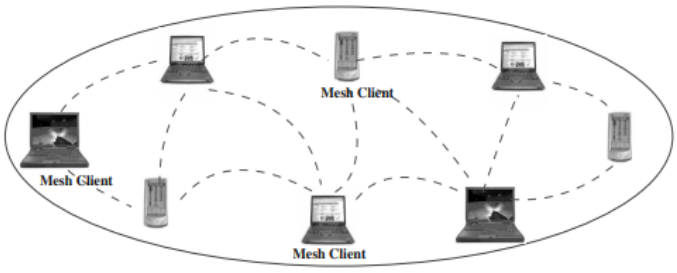

Figura 2. Cliente mesh

Mesh híbrido: Esta arquitectura combina la infraestructura con los clientes mesh. Los clientes mesh podrían acceder a la red a través de la red de routers o a través de otros clientes mesh aumentando así la cobertura. Además de ello, se interconectan los otros tipos de redes ya

existentes como puede ser Wifi, WiMax, redes móviles, radio,... Los routers mesh tienen una movilidad reducida y están concentrados en realizar todas las tareas de encaminamiento y configuración facilitando la tarea de los clientes y otros nodos y reduciendo su trabajo. Se mantiene la tecnología multi-hop gracias a la red de routers desde la que no es necesario que todos los nodos tengan completa visión de todos los nodos existentes, sino que tan sólo es necesario visualizar los nodos cercanos.

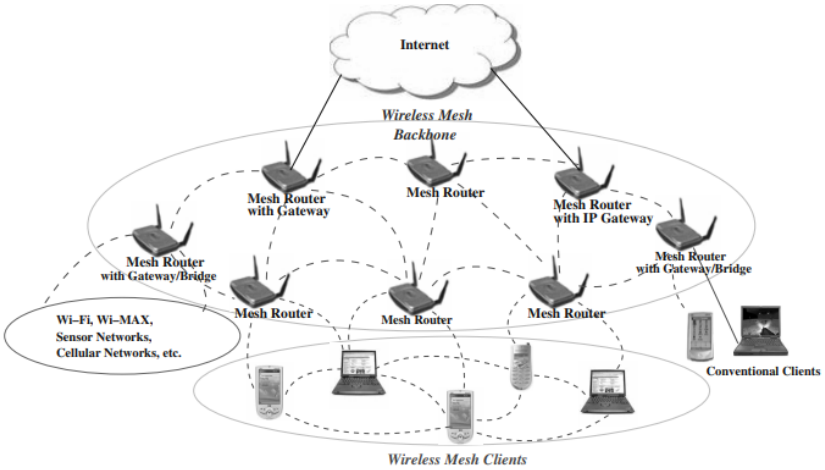

Figura 3. Mesh híbrido

4.1.2. Protocolos para WMN

Dado que se trata de una red de topología dinámica y auto configurable, las rutas que se establecen entre dispositivos cambian dinámicamente, con lo cual son necesarios una serie de protocolos para transmitir los datos con un bajo coste de transmisión.

Existen una serie de protocolos de enrutamiento divididos en dos clases: proactivos y reactivos. Los proactivos mantienen actualizadas sus tablas de enrutamiento en todo momento y la comunicación se establece prácticamente al instante. Los reactivos en cambio, se ejecutan tras una demanda de datos y, por tanto, se requiere un cálculo inicial antes de que se establezca la comunicación.

A continuación, presentamos un listado de los protocolos más destacados.

OLSR (Optimized Link State Routing Protocol): Es un protocolo de enrutamiento por IP para redes móviles o inalámbricas ad-hoc. Es un protocolo proactivo, que envía de forma distribuida mensajes de “Hello” para conocer los nodos a su alcance y una vez los tiene, envía

mensajes de TC (Topology Control) a un subconjunto de estos para establecer las conexiones. El OLSR-ETX usa como criterio el número de pérdidas de una ruta, seleccionando así las de mayor calidad de enlace.

MMRP (MobileMesh): Este protocolo, también proactivo, y contiene tres protocolos separados, cada uno destinado a una función específica:

Mobile Mesh Link Discovery Protocol (MMLDP): descubrir los enlaces disponibles, con un mensaje "hello".

Mobile Mesh Routing Protocol (MMRP): protocolo de verificación de estado de enlaces para enrutamiento.

Mobile Mesh Border Discovery Protocol (MMBDP): descubre bordes y habilita túneles externos para conectar con otras redes.

AODV (Ad-hoc On Demand Distance Vector): Como su propio nombre indica, es un protocolo de enrutamiento de vector distancia. Es un protocolo reactivo, por lo que la tabla de enrutamiento sólo se actualiza tras una demanda y la información recuperada permanece almacenada el tiempo necesario para que se realice la comunicación. Cuando un nodo demanda información, envía mensajes de "route request" (RREQ) y espera a que los nodos adyacentes contesten con un "route reply" (RREP) para formar la ruta. Una vez creada la ruta, si uno de los nodos falla, se envía un error (RERR) al nodo que demanda y vuelve a buscar una ruta óptima.

HSLS (Hazy Sighted Link State Routing Protocol): Se trata de un protocolo proactivo a la par que reactivo para limitar las actualizaciones de enrutamiento en espacio y tiempo. Se diseñó para operar en redes de más de mil nodos. Consiste en desechar los enlaces de baja calidad.

OSPF (Open Shortest Path First): En este protocolo proactivo los nodos envían llamadas, verifican el estado de los enlaces y transmiten la información recopilada a todos los enrutadores de una misma área jerárquica. Los enrutadores calculan el camino más corto usando el algoritmo SPF (Shortest Path First), el cual genera un árbol seleccionando siempre la ruta más corta, y almacenan la información. Este protocolo además funciona como LSA(Link-State Advertisement) y avisa a las interfaces presentes, informa del tipo de medición usada y otras variables.

TBRPF (Topology Broadcast based on Reverse PathForwarding): Este protocolo proactivo escoge la ruta más corta en saltos generando un árbol por cada nodo que se calcula mediante una modificación del algoritmo de Dijkstra. Para minimizar la sobrecarga en la red, cada nodo enviará información a sus vecinos de un subconjunto de nodos de su árbol.

4.2. Estándar 802.11s (Jerome, 2011)

Es un estándar de red inalámbrico y una enmienda al protocolo IEEE 802.11 para las redes malladas, que define como los dispositivos inalámbricos pueden interconectarse para crear una red inalámbrica en malla, que puede utilizarse para topologías relativamente fijas (no móviles) y redes inalámbricas ad hoc.

Los dispositivos de dichas redes tienen la inteligencia suficiente como para ir creando ellos mismos las rutas de manera dinámica optimizando el tráfico de información y evitando fallos en la conectividad en caso de que determinados nodos de la red se desconecten.

Encontrar un camino para llegar a un destino determinado se denomina Selección de Ruta en 802.11s. Este destino suele ser la red cableada, pero también puede ser cualquier otra dirección MAC alcanzable a través de la MBSS (Mesh Basic Service Set)

Las mesh stations que implementen 802.11s también deben ser compatibles con el protocolo de selección de rutas predeterminado, Hybrid Wireless Mesh Protocol (HWMP). HWMP proporciona selección de ruta proactiva y selección de ruta reactiva. Una mesh station que necesite transmitir una trama a un destino desconocido puede descubrir dinámicamente la mejor ruta a este destino. Las estaciones de malla también pueden descubrir proactivamente la MBSS y determinar las mejores rutas a cualquier punto de la nube de malla antes de necesitar enviar cualquier trama de datos.

4.2.1. HWMP

Es el protocolo de rutado que especifica el nuevo estándar IEEE 802.11s para redes malladas, es un protocolo híbrido que tiene un conocimiento parcial de la topología, es decir conoce sólo a los vecinos. Este protocolo debe ser implementado obligatoriamente por todos los nodos mesh, aunque se permite usar protocolos adicionales. La principal ventaja de este estándar es que introduce un mecanismo de enrutamiento en la capa 2 (MAC), haciéndolo aparecer como un sistema LAN (802.x) para protocolos de capas superiores. Además, define

no sólo cuestiones de encaminamiento sino también aspectos como acceso al medio, sincronización o seguridad. El hecho de que el enrutamiento funcione en la capa de enlace de datos también se convierte en una desventaja, ya que de esta manera no se puede aprovechar la estructura jerárquica de protocolos de direccionamiento superiores, como IP, ni interconectar diferentes redes. Este hecho hace que sea complicado enrutar paquetes sólo con HWMP en redes mesh de tamaño medio o grande. Por ello se hace necesario combinar este protocolo con otros de capas superiores.

HWMP combina características del protocolo AODV y técnicas de enrutamiento basadas en árbol. La combinación de elementos proactivos y reactivos permite una óptima y eficiente selección de ruta en una amplia variedad de redes mesh (con y sin infraestructura). Utiliza un conjunto de mensajes basados en AODV, adaptados al direccionamiento MAC de la capa 2, para el descubrimiento de rutas reactivamente.

Adicionalmente, se usan otros mensajes para, de forma proactiva, construir un árbol de vectores distancia a partir de un nodo raíz. Este último método necesita que el nodo a partir del cual se calculará el árbol esté configurado como raíz. Pueden existir varios nodos raíz en una misma red. Esta técnica suele utilizarse para construir árboles de rutas hacia nodos que cumplen un papel especial dentro de la red. Se contemplan dos modos de funcionamiento, no excluyentes: bajo demanda y construcción proactiva de árbol. Este modo híbrido permite combinar concurrentemente componentes reactivos y proactivos. Permite elegir cualquier métrica o combinación de métricas.

Este protocolo combina la flexibilidad de la sobre-demanda de descubrimiento de ruta con eficiente enrutamiento proactivo hasta un portal mesh, con características como:

- El enrutamiento bajo demanda ofrece gran flexibilidad en ambientes variantes.
- El enrutamiento basado en un árbol pro-activo es muy eficiente en el despliegue de MESH fijas.
- La combinación lo hace adecuado para la implementación sobre una variedad de dispositivos bajo las consideraciones en modelos usados por la TGs (Task Groups)

4.2.2. Métrica ALM (Airtime Link Metric)

Estima la cantidad de recursos que se utilizan en el canal, tomando en cuenta factores como la tasa de pérdidas de paquetes, el ancho de banda.

$$C_a = (O_{ca} + O_p + \frac{B_t}{r}) * (\frac{1}{1 - e_{pt}})$$

	802.11a	**802.11 b/g**	**Descripción**
O_{ca}	75 us	335 us	Sobrecarga del acceso al canal
O_p	110 us	364 us	Sobrecarga del protocolo
B_t	8192	8192	Bits en la trama de prueba

Tabla 1. Métrica AML

4.3. Simulador NS-3

El simulador de redes 3 (NS-3, por sus siglas en inglés) es una herramienta que se emplea para simular eventos discretos de una red, especialmente si se necesita simular escenarios en las capas 2 y 4 del modelo de interconexión de sistemas abiertos.

NS-3 presenta las siguientes características que lo distinguen de los otros simuladores:

- Diseñado como un conjunto de bibliotecas que se pueden combinar entre sí y también con otras bibliotecas de software externos.
- Es modular, presenta varios animadores externos para el análisis de datos y herramientas de visualización.
- Desarrollo en software C ++ y / o Python.
- Mayormente utilizado en sistemas Linux, existe soporte para FreeBSD, Cygwin (para Windows), y se encuentra en desarrollo el apoyo de Visual Studio nativo de Windows.

El proyecto NS-3 se ha comprometido a construir un núcleo de simulación sólido, bien documentado, fácil de usar y depurar, y que satisfaga las necesidades de todo el flujo de

trabajo de simulación, desde la configuración de la simulación hasta la recopilación y el análisis de trazas.

Además, la infraestructura de software de NS-3 fomenta el desarrollo de modelos de simulación lo suficientemente realistas como para que NS-3 pueda utilizarse como emulador de redes en tiempo real, interconectado con el mundo real y que permita reutilizar en NS-3 muchas implementaciones de protocolos existentes en el mundo.

Los elementos básicos de una simulación vienen representados por clases C++ que simulan los elementos básicos hardware/software de un escenario real, son los siguientes:

1. Node: representa un dispositivo (computador, servidor, teléfono móvil, portátil, sensor, cámara, etc.)
2. Channel: representa un canal de comunicación y su comportamiento (cable, inalámbrico, punto a punto, etc.)
3. Net Device : Dispositivo de red, es una tarjeta de red inalámbrica (e.j wifi) o cableada (e.j ethernet)
4. Application: es una aplicación software. Desde el punto de vista de una simulación es un generador/consumidor de paquetes de información. Un servidor web, un navegador, etc. son ejemplos de aplicaciones
5. Protocol stack: es una pila de protocolos de comunicación. El más conocido, la pila de protocolos TCP/IP que hace posible la comunicación en Internet
6. El simulador propiamente dicho: se encarga de lanzar y gestionar la simulación.

Con estos cinco elementos podemos simular cualquier escenario de red. En nuestro archivo de C++ debemos tener estos elementos configurados apropiadamente para simular un escenario real.

4.3.1. 802.11s en NS-3

Figura 4. Red mallada nxm

La Figura 4 muestra la configuración de una red mallada nxm donde n y m son el número de nodo por el que se encuentra formada la malla, n representa los nodos ubicados de manera horizontal y m representa los nodos ubicados de manera vertical, dichos nodos se encuentran separados entre sí una distancia step. En este caso se hace ping desde el primer nodo al último nodo.

1. Se debe cargar los archivos de inclusión.
2. Luego de los módulos de inclusión sigue el espacio de nombres ns-3 el mismo que contiene las declaraciones de ns-3.
3. Se define el componente de registro.
4. Se declaran las variables empleadas en el script.
5. Se establece el valor inicial de las variables declaradas anteriormente, esto se lo realiza dentro del programa principal.
6. Se crean el número de nodos en forma de malla, que se crea necesario los mismos que representaran los Nodos.
7. Se habilita los Logs.
8. Se emplea un ayudante para crear y administrar los objetos PHY.
9. Modificar los parámetros del canal.
10. Se crea un ayudante de malla, instala las pilas y crea todos los protocolos necesarios e instala en los dispositivos de punto de malla.

11. Se establece el número de interfaces, el valor predeterminado es el punto de malla de una sola interfaz.
12. Se instala los protocolos mesh.
13. Se modifica la movilidad de los nodos con el ayudante, a una topología de red estática.
14. Se procede a crear la pila de protocolos
15. Se asigna las direcciones IP a las interfaces de los dispositivos.
16. Se crea el nivel de aplicaciones.
17. Se habilita el rastreo.
18. Se establece un contador para que se imprima un reporte según la secuencia deseada.
19. Se activa el reporte para cada uno de los MP.
20. Se ejecuta la simulación.

IEEE 802.11s [2] es el estándar de facto para las implementaciones de WMN. Este estándar define dos nuevos protocolos: el protocolo híbrido de malla inalámbrica (HWMP) y el protocolo de gestión de pares (PMP) para soportar la funcionalidad de malla inalámbrica. El protocolo HWMP proporciona la funcionalidad de enrutamiento de capa 2, mientras que el PMP ayuda a mantener los enlaces entre los puntos de malla.

4.4. Modelos lineales.

El análisis de regresión es un proceso estadístico que permite analizar la relación que existe entre dos o más variables, siendo una de ellas dependiente al resto de variables que estemos empleando en nuestro cálculo matemático. Por lo que podemos decir que un análisis regresivo hace posible comprender cómo las variables independientes afectan directamente a otra variable que depende de ellas.
Las variables se dividen en dos grupos:

- **Variables dependientes (respuesta):** son aquellas que buscamos estudiar mediante la regresión estadística para comprender cómo se adapta al modificar las variables independientes.

- **Variables independientes (predictivos):** Son los factores que consideramos que influyen y que afectan directamente a las variables dependientes que están bajo estudio.

4.4.1. El modelo lineal

La regresión lineal (modelo lineal) es el modelo más común y se supone que es la relación entre dos variables tiene una forma lineal (o linealizable mediante alguna transformación de las variables). La regresión lineal tiene una versión "simple" que empareja dos variables (dependiente e independiente), pero esta suele ser insuficiente para entender fenómenos mínimamente complejos en la que influyen más de dos variables.

Los modelos de regresión lineal pueden ser:

- **Regresión Lineal Simple:** es el más utilizado y el más sencillo de todos. Se trata de estudiar el efecto de una variable independiente sobre una única variable dependiente de la primera.

$$Y = \beta 0 + \beta i X + \epsilon i$$

Donde β representa las estimaciones de parámetros lineales que se deben de calcular y ϵ representa los términos de error.

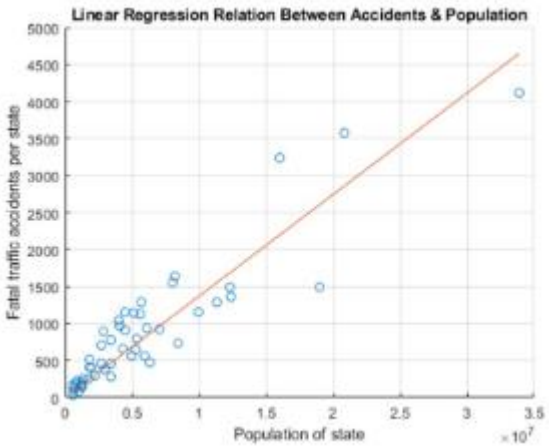

Figura 5. Regresión lineal simple.

- **Regresión Lineal Multiple:** Utilizan múltiples predictores. esta regresión tiene múltiples variables independientes (X), para predecir la respuesta (Y).

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \epsilon$$

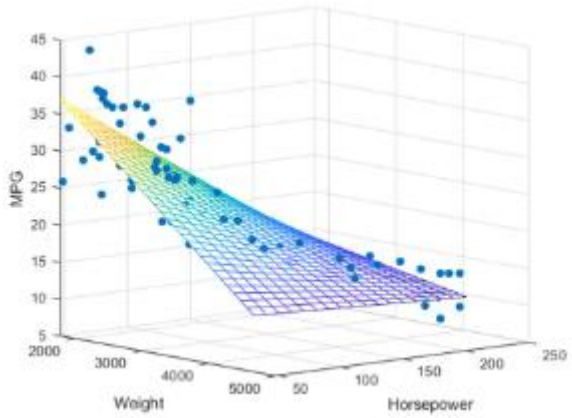

Figura 6. Regresión lineal múltiple.

- **Regresión Lineal Multivariante:** modelos para varias variables de respuesta. Esta regresión tiene múltiples Yi que derivan de los mismos datos Y. Se expresan con fórmulas diferentes.

$$Y1 = \beta 01 + \beta 11X1 + \epsilon 1$$

$$Y2 = \beta 02 + \beta 12X1 + \epsilon 2$$

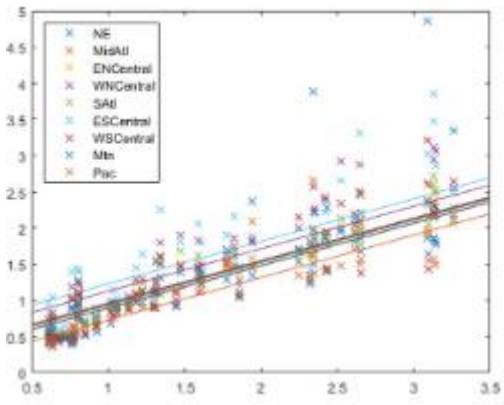

Figura 7. Regresión lineal multivariable

El modelo lineal utilizado para el análisis de los datos es el modelo lineal multivariante la cual es utilizada para varias variables de respuesta. Esta regresión tiene múltiples Yi que derivan de los mismos datos Y

4.4.2. Análisis del modelo lineal

Para iniciar con el análisis del modelo lineal se debe considerar que existe una relación lineal entre las variables dependientes e independientes como se puede observar en el siguiente gráfico.

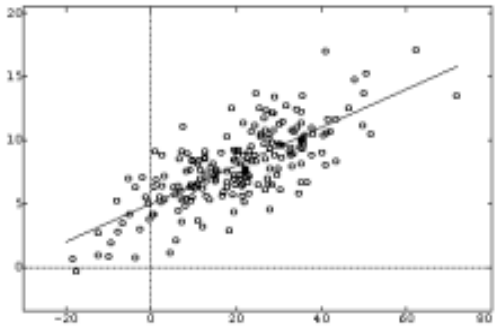

Figura 8. Regresión lineal simple

Como se puede observar en el gráfico anterior la relación entre y y x es lineal y se representa con la siguiente ecuación:

$$yi = \alpha + \beta xi + \varepsilon i$$

No todos valores de x se encuentran a lo largo de la recta con lo que se admite un error de medición ε, lo que sería una variabilidad individual de las unidades estadísticas.

Por tal motivo se puede decir que la elección del modelo se lo hace por:

1. Ya se sabe que la relación es lineal.
2. En la región de las elecciones usuales de x la relación lineal es muy buena aproximación,
3. Se busca una relación funcional entre y y x y se utiliza un modelo lineal como primera etapa de investigación.

4.4.3. Estimación de los parámetros

Hay que buscar un método que permita estimar los parámetros desconocidos α y β para encontrar la mejor recta en algún sentido.

Entonces se busca la recta que sea más cercana a los datos, esto se puede realizar minimizando su distancia desde cada punto.

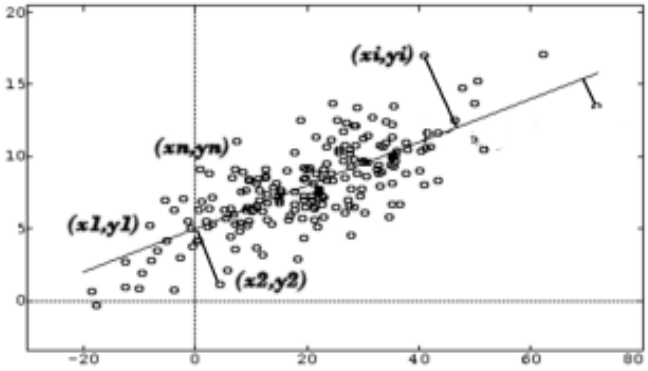

Figura 9. Estimación de parámetros

por cada observación de los puntos (xi, yi) se tiene ε con respecto a la recta. Hay tres posibles determinaciones de error:

1. $\sum i\ \varepsilon i$, pero las diferencias de signal no permiten optimizar;
2. $\sum i\ |\varepsilon i|$ pero el algebra de los valores absolutos es difícil;
3. $\sum i \varepsilon_i^2$ mejor, pero pesos grandes para errores grandes y reducidos para pequeños

La recta se puede ajustar minimizando su distancia desde cada punto.

4.4.4. La recta de los mínimos cuadrados

Para identificar la recta de regresión se utilizará la que minimiza la suma de cuadrados de los errores, este valor corresponde también a la calidad de la estimación.

Así que, entre la infinidad de rectas de regresión posibles, se elegirá la estimación de los parámetros.

$$SSe(\hat{}\alpha, \beta\hat{}) = m\acute{}\imath n\ (\alpha, \beta)\ SSe(\alpha, \beta) = m\acute{}\imath n\ (\alpha, \beta) \sum (yi - \alpha - \beta xi)\ 2$$

4.4.5. Contribución y apalancamiento

Vemos ahora como estudiar la contribución de las observaciones (xi, yi) a la determinación de la recta de regresión. Como se sabe, esta pasa para el baricentro ($\underline{x}$, $\underline{y}$) de la nube de puntos y es enteramente determinata para su pendiente $\hat{\beta}$, cuya determinación se puede escribir:

$$\hat{\beta} = \frac{\Sigma_i (y_i - \bar{y})(x_i - \bar{x})}{\Sigma_i \ (x_i - \bar{x})^2} = \sum_{i=1}^{n} \left(\frac{(x_i - \bar{x})^2}{\Sigma_l (x_l - \bar{x})^2} \right) \frac{y_i - \bar{y}}{x_i - \bar{x}} = \sum_i w_i b_i$$

Como bi = $\frac{y_i - \underline{y}}{x_i - \underline{x}}$ es un pendiente, $\hat{\beta}$ es el promedio pesado de los pendientes de las rectas que unen cada (x_i , y_i) con ($\underline{x}$,) con peso wi proporcional al cuadrado de su distancia de xi de ¯x.

4.5. RStudio

4.5.1. Conceptos fundamentales

R Software es la herramienta líder en análisis estadístico, manipulación de grandes volúmenes de datos y representación gráfica de alta calidad. Además, es un software de código abierto y disponible para todos los sistemas operativos Windows, MACOs, Unix y Linux.

RStudio es un programa que nos permite interactuar con R de forma más amigable y versátil, además de facilitar muchas de las tareas de programación y análisis de datos en R; es decir, RStudio es una GUI (Graphical User Interface).

4.5.2. El modelo lineal en RStudio

Los modelos lineales son una de las herramientas más importantes del análisis cuantitativo. Se utilizan cuando se desea predecir o explicar una variable dependiente a partir de una o más

variables independientes. El ajuste de modelos lineales busca encontrar una función matemática que permita expresar a una variable a partir de otras.

4.5.3. Ajuste de modelo lineales con lm ()

REGRESIÓN LINEAL SIMPLE: La regresión lineal simple supone que los valores de la variable dependiente, a los que llamaremos yi, pueden escribirse en función de los valores de una única variable independiente, los cuales notaremos por xi, según el siguiente modelo lineal:

$$y_i = \beta_O \ + \beta_1 \, x_i$$

donde $\beta_0 \ y \ \beta_1$ son los parámetros desconocidos que vamos a estimar.

Habitualmente, al iniciar un estudio de regresión lineal simple se suelen representar los valores de la variable dependiente y de la variable independiente de forma conjunta mediante un diagrama de dispersión para determinar si realmente existe una relación lineal entre ambas. Para realizar un diagrama de dispersión en R y RStudio utilizaremos la orden plot

> plot(x,y)

donde x e y son los valores de las variables independiente y dependiente, respectivamente. En caso de que en el diagrama de dispersión se aprecia un patrón lineal entre las dos variables, se podrá asumir una cierta relación lineal entre ambas variables y se procederá a ajustar el modelo de regresión lineal simple.

Después de comprobar gráficamente la relación lineal entre las variables, el siguiente paso es la estimación de los valores de los parámetros β_0 y β_1 que aparecen en la fórmula (1) a partir de un conjunto de datos. Para ello, podemos utilizar la función lm de R, cuya sintaxis es la siguiente:

> lm(formula, data)

donde la fórmula indica la relación que guardan la variable dependiente y la variable independiente.

Por ejemplo:

> lm(formula = y ~ x, data=midataset)

Algunas peculiaridades sobre este argumento:

- La parte izquierda (variable dependiente) y derecha (variable independiente) de la fórmula vienen separadas por el símbolo **~**.

- ***data*** es el conjunto de datos en el que se encuentran las variables que se utilizan en la fórmula. Este argumento es optativo, aunque es muy recomendable usarlo para una mayor claridad en la fórmula.

REGRESION LINEAL MULTIPLE: El modelo de regresión múltiple es la extensión a *k* variables explicativas del modelo de regresión simple. En general, una variable de interés y depende de varias variables *x1, x2, …, xk* y no sólo de una única variable de predicción x.

Un modelo de regresión representa el efecto de estas variables en lo que se conoce como error aleatorio o perturbación.

Un modelo de regresión teórico en el que las variables se pueden relacionar mediante una función de tipo lineal, podemos expresarlo de la siguiente forma:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

donde

- ***y*** es la variable de interés que vamos a predecir, también llamada variable respuesta o variable dependiente
- *x1, x2, …, xk* son variables independientes, explicativas o de predicción
- β1, β2, …, βk son los parámetros desconocidos que vamos a estimar
- ε es el error aleatorio o perturbación, que representa el efecto de todas las variables que pueden afectar a la variable dependiente y no están incluidas en el modelo de regresión.

En ***R y Rstudio***, el paso de un modelo de regresión lineal simple a un modelo de regresión lineal múltiple es muy sencillo: basta con añadir variables independientes al argumento ***formula*** de la función ***lm*** separadas por el signo **+**.

lm(dependiente~independiente1+independiente2, data=datos)

CORRELACION LINEAL: Para calcular el coeficiente de correlación lineal entre dos variables en **R y RStudio** utilizan la orden **cor**, cuya sintaxis es la siguiente:

cor (x, y, method = c("pearson" "kendall", "spearman"))

donde

- **x** hace referencia a la primera de las variables
- **y** hace referencia a la segunda de las variables
- **method** indica el método que se va a utilizar para calcular el coeficiente de correlación lineal.

5. Metodología

5.1. Descripción de los scripts en NS-3

5.1.1. Mesh-ping

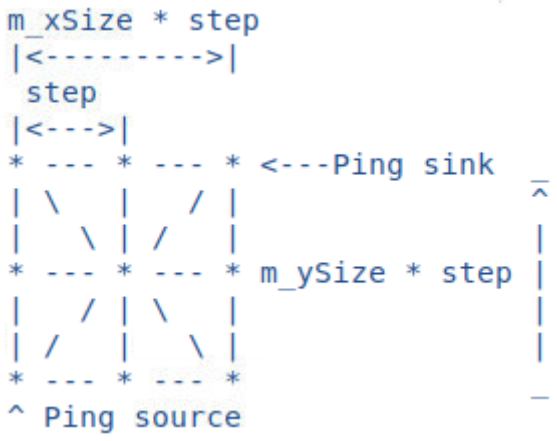

Figura 10. Mesh-ping

Este script crea una topología de una red mallada cuadrada de tamaño m_x * m_y, siendo el número de nodos por el cual se encuentra formada la malla, los nodos se encuentran separados por el parámetro step.

Cuando la topología es creada UDP ping es instalada en las esquinas opuestas de la diagonal, el tamaño del paquete UDP ping y el intervalo entre los dos paquetes sucesivos es configurable.

A continuación, se presenta la estructura del programa:

- *Declaración de librerías*

```
#include <iostream>
#include <sstream>
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mesh-module.h"
#include "ns3/mobility-module.h"
#include "ns3/mesh-helper.h"
#include "ns3/yans-wifi-helper.h"
```

Figura 11. Declaración de librerías.

- *Declaración de variables*

```
int        m_xSize; ///< X size
int        m_ySize; ///< Y size
int aux;    f
double     m_step; ///< step
double     m_randomStart; ///< random start
double     m_inicio;
double     m_totalTime; ///< total time
double     m_packetInterval; ///< packet interval
uint16_t   m_packetSize; ///< packet size
uint32_t   m_nIfaces; ///< number interfaces
bool       m_chan; ///< channel
bool       m_pcap; ///< PCAP
bool       m_ascii; ///< ASCII
std::string m_stack; ///< stack
std::string m_root; ///< root
```

Figura 12. Declaración de variables.

- *Inicialización de variables*

```
cmd.AddValue ("x-size", "Number of nodes in a row grid", m_xSize);
cmd.AddValue ("y-size", "Number of rows in a grid", m_ySize);
cmd.AddValue ("step",   "Size of edge in our grid (meters)", m_step);
// Avoid starting all mesh nodes at the same time (beacons may collide)
cmd.AddValue ("start",  "Maximum random start delay for beacon jitter (sec)", m_randomStart);
cmd.AddValue ("time",  "Simulation time (sec)", m_totalTime);
cmd.AddValue ("packet-interval",  "Interval between packets in UDP ping (sec)", m_packetInterval);
cmd.AddValue ("packet-size",  "Size of packets in UDP ping (bytes)", m_packetSize);
cmd.AddValue ("interfaces", "Number of radio interfaces used by each mesh point", m_nIfaces);
cmd.AddValue ("channels",   "Use different frequency channels for different interfaces", m_chan);
cmd.AddValue ("pcap",   "Enable PCAP traces on interfaces", m_pcap);
cmd.AddValue ("ascii",   "Enable Ascii traces on interfaces", m_ascii);
cmd.AddValue ("stack",  "Type of protocol stack. ns3::Dot11sStack by default", m_stack);
cmd.AddValue ("root", "Mac address of root mesh point in HWMP", m_root);
```

Figura 13. Inicialización de variables

En la Figura 13. se definen los parámetros de una simulación de red Mesh inalámbrica, de esta manera se tiene:

- El tamaño de la red o número de nodos en un arreglo en forma de malla (Línea 1 - 2)
- La distancia física entre nodos definida en metros (Línea 3)
- La inicialización de un retardo para evitar colisiones iniciales de paquetes (Línea 5)
- La duración de la simulación (Línea 6)
- La configuración de tráfico de prueba (Línea 7 - 8)
- El número de interfaces de radio por nodo (Línea 9)
- Canales diferentes para evitar interferencias (Línea 10)
- Generación de archivos PCAP para análisis (Línea 11)
- Generación de trazas ASCII para depuración (Línea 12)
- Tipo de protocolo (Línea 13)
- Dirección MAC del nodo raíz en HWMP (Línea 14)

Configuración de enrutamiento, para el caso del trabajo desarrollado

- *Creación de la red mallada*

nodes.Create (m_ySize*m_xSize)

Se crean los nodos, los cuales son entidades abstractas en ns-3 y se pueden equipar con stacks de protocolos, interfaces de red y modelos de movilidad.

- *Inicialización de un ayudante de configuración predefinida para redes Mesh*

mesh = MeshHelper::Default ()

Los parámetros configurados por default se detallan a continuación:

Parámetro	Valor / Clase Predeterminada	Descripción
Protocolo de Ruteo	ns3::Dot11sStack	Implementación de 802.11s con HWMP (Hybrid Wireless Mesh Protocol).
Métrica de Enlace	ns3::AirtimeLinkMetricCalculator	Calcula el costo de ruta basado en retardo y tasa de error.
Beacon Interval	1 segundo	Frecuencia de mensajes de beacon para mantenimiento de la topología.
Número de Interfaces	1	Interfaces WiFi por nodo (puede modificarse con SetNumberOfInterfaces()).

Tabla 2. Configuración de parámetros de enrutamiento.

- *Establecer la posición de los nodos en el plano x, y en función de la variable distancia.*

```
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                               "MinX", DoubleValue (0.0),
                               "MinY", DoubleValue (0.0),
                               "DeltaX", DoubleValue (m_step),
                               "DeltaY", DoubleValue (m_step),
                               "GridWidth", UintegerValue (m_xSize),
                               "LayoutType", StringValue ("RowFirst"));
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodes);
```

Figura 14. Posición de nodos.

El código mostrado configura el posicionamiento y movilidad de nodos en una simulación de ns-3 para una topología de red en rejilla estática.

- *Activación de reporte*

```
for (Seconds(m_inicio);(Seconds(m_inicio) < Seconds(m_totalTime));m_inicio++)
{
      Simulator::Schedule (Seconds (m_inicio), &MeshTest::Report, this, aux);

       aux=m_inicio;
}
```

Figura 15. Activación del reporte.

El código muestra un bucle de programación de eventos donde se agenda periódicamente la ejecución de un método para generar reportes durante la simulación.

- *Ejecutar la simulación.*

5.1.2. Airtime-Metric.

El airtime-métric donde las métricas separadas del enlace se agregan hasta conseguir la trayectoria métrica.

A continuación, se presenta la estructura del programa:

- *Declaración de librerías*

```
#include "airtime-metric.h"
#include "ns3/wifi-phy.h"
```

Figura 16. Declaración de librerías Airtime-Metric.

- *Parametrización del Objeto AirtimeLinkMetricCalculator*

```
TypeId
AirtimeLinkMetricCalculator::GetTypeId ()
{
  static TypeId tid = TypeId ("ns3::dot11s::AirtimeLinkMetricCalculator")
    .SetParent<Object> ()
    .SetGroupName ("Mesh")
    .AddConstructor<AirtimeLinkMetricCalculator> ()
    .AddAttribute ( "TestLength",
                    "Number of bytes in test frame (a constant 1024 in the standard)",
                    UintegerValue (1024),
                    MakeUintegerAccessor (
                      &AirtimeLinkMetricCalculator::SetTestLength),
                    MakeUintegerChecker<uint16_t> (1)
                    )
    .AddAttribute ( "Dot11MetricTid",
                    "TID used to calculate metric (data rate)",
                    UintegerValue (0),
                    MakeUintegerAccessor (
                      &AirtimeLinkMetricCalculator::SetHeaderTid),
                    MakeUintegerChecker<uint8_t> (0)
                    )
  ;
  return tid;
```

Figura 17. Parametrización Objeto AirtimeLinkMetric

El código personaliza el comportamiento del protocolo HWMP en redes mesh, permitiendo ajustar la manera de evaluación en la calidad de los enlaces inalámbricos.

- *Cálculo*

```
  NS_ASSERT (!peerAddress.IsGroup ());
  //obtain current rate:
  WifiMode mode = mac->GetWifiRemoteStationManager ()->GetDataTxVector (peerAddress,
&m_testHeader, m_testFrame).GetMode();
  //obtain frame error rate:
  double failAvg = mac->GetWifiRemoteStationManager ()->GetInfo
(peerAddress).GetFrameErrorRate ();
  if (failAvg == 1)
    {
      // Return max metric value when frame error rate equals to 1
      return (uint32_t) 0xffffffff;
    }
  NS_ASSERT (failAvg < 1.0);
  WifiTxVector txVector;
  txVector.SetMode (mode);
  txVector.SetPreambleType (WIFI_PREAMBLE_LONG);
  //calculate metric
  uint32_t metric = (uint32_t)((double)( /*Overhead + payload*/
                                mac->GetPifs () + mac->GetSlot () + mac->GetEifsNoDifs ()
+ //DIFS + SIFS + AckTxTime = PIFS + SLOT + EifsNoDifs
                                mac->GetWifiPhy ()->CalculateTxDuration (m_testFrame-
>GetSize (), txVector, mac->GetWifiPhy ()->GetFrequency())
                                ).GetMicroSeconds () / (10.24 * (1.0 - failAvg)));
  return metric;
```

Figura 18. Cálculo

El código implementa el cálculo de la métrica de airtime para enlaces inalámbricos en redes mesh 802.11s, basado en el tiempo de transmisión, la tasa de error del enlace y los overheads de protocolo.

6. Resultados.

6.1. Descripción de Script en RStudio

- *Carga de librerías*

Se procede a cargar las librerías detalladas a continuación:

Librería	Uso	Librería	Uso
lmtest	Pruebas para modelos lineales (heterocedasticidad, autocorrelación)	sandwich	Errores estándar robustos para modelos
pracma	Matemáticas prácticas (álgebra lineal, optimización)	lattice	Sistema de gráficos avanzados
lrmest	Estimación robusta en modelos lineales	viridisLite	Escalas de color para visualización

Librería	Uso	Librería	Uso
corrplot	Visualización de matrices de correlación	leaps	Selección de modelos (métodos forward/backward)
tseries	Análisis de series temporales (pruebas de estacionariedad)	agricolae	Diseños experimentales y análisis agrícolas
nortest	Pruebas de normalidad (Anderson-Darling, Shapiro-Wilk)	faraway	Conjuntos de datos y funciones para regresión
car	Análisis de regresión (VIF, pruebas de hipótesis)	stargazer	Presentación profesional de resultados (tablas LaTeX/HTML)
dplyr	Manipulación avanzada de datos (sintaxis tidy)	MASS	Funciones estadísticas avanzadas (regresión robusta, LDA)

Tabla 3. Librerías utilizadas en RStudio.

```
suppressPackageStartupMessages({
  suppressWarnings({
    suppressMessages({
      library(lmtest)
      library(pracma)
      library(lrmest)
      library(corrplot)
      library(tseries)
      library(nortest)
      library(car)
      library(sandwich)
      library(lattice)
      library(viridisLite)
      library(leaps)
      library(agricolae)
      library(faraway)
      library(stargazer)
      library(dplyr)
      library(MASS)

    })
  })
})
```

Figura 19. Carga de librerías

- *Carga de datos*

```
27  mesh2 <- read.csv("mesh_udp_2.csv", header = TRUE, sep = ",", quote = "\"", dec =
        ".", fill = TRUE, comment.char = "", encoding = "unknown")
```

Figura 20. Carga de datos

- *Reducción de muestras a 1000*

```
32  set.seed(123)
33  dt_index <- 1:nrow(mesh2)
34  smpl_index <- sample(dt_index, size = floor(0.021 * length(dt_index)), replace =
        FALSE)
35  mesh_fit <- mesh2[smpl_index, ]
```

Figura 21. Reducción de Muestras

Este código realiza un muestreo aleatorio para crear una submuestra más pequeña considerando:

- La creación de un vector de índices a muestrear
- El cálculo del 2.1% del total de filas
- El muestreo sin reemplazo
- La creación del nuevo dataframe con las filas seleccionadas aleatoriamente

- *Ejecución del programa*

```
#Inicio del programa
attach(mesh_fit)
n   <- dim(mesh_fit)[1]                    # n mero de unidades
p   <- dim(mesh_fit)[2]                    # n mero de variables
lab <- rownames(mesh_fit)                  # etiquetas de las unidades en lab
nom <- colnames(mesh_fit)                  # etiquetas de las variables en nom
# chequeo de los datos iniciales
print(head(mesh_fit, 10))                  # encabezamiento
print(str(mesh_fit))                       # structura del data.frame
print(summary(mesh_fit))                   # estad sticas som rias
# Conocer las variables categorizantes
table(as.factor(MXN))
table(as.factor(TXDURATION))
table(as.factor(M_STEP))
table(as.factor(M_PACKET))
table(as.factor(INTERFACES))
plot( ..FAILAVG,METRIC)
# diagramas de caja y bigotes entre un caracter cuantitativo y cualitativos
boxplot(METRIC~MXN)
boxplot(METRIC~TXDURATION)
boxplot(METRIC~M_STEP)
boxplot(METRIC~M_PACKET)
boxplot(METRIC~INTERFACES)
```

Figura 22. Ejecución del programa

El código presentado en la Figura 22, realiza las siguientes acciones:

- Proporciona información básica sobre la estructura de los datos.
- Prepara identificadores para las observaciones.
- Crea objetos que servirán para generar gráficos con etiquetas adecuadas y referenciar variables de manera programada.
- Muestra las 10 primeras filas del dataframe.
- Muestra la estructura interna del Objeto.
- Calcula estadísticas resumidas para cada columna
- Analiza variables para realizar una distribución de categorías.
- Relaciona variables numéricas.
- Genera diagramas para visualizar la distribución de la variable segmentada por diferentes variables categóricas.

- *Conversión a factores*

```
##1. Modelo lineal con metric y agregar regresores
#Definimos regresores como factor
M_STEP=factor(M_STEP)
M_PACKET=factor(M_PACKET)
INTERFACES=factor(INTERFACES)
MXN=factor(MXN)
TXDURATION=factor(TXDURATION)
```

Figura 23. Conversión a factores

El código convierte cinco variables del dataframe (M_STEP, M_PACKET, INTERFACES, MXN, TXDURATION) en factores, que es el tipo de dato adecuado para variables categóricas en R.

6.1.1. Modelo 1. Modelo de Regresión lineal

```
##Modelo 1 IEEE 802.11s Original
lmesh1  <- lm(METRIC~ ..FAILAVG, data=mesh_fit)
summary(lmesh1)
anova(lmesh1)
par(mfcol = c(2, 2))
plot(lmesh1)
boxplot(lmesh1[['residuals']],main='Boxplot: Residuals',ylab='residual value')
```

Figura 24. Modelo de regresión lineal

El código realiza un modelo de regresión lineal para predecir METRIC en función de FAILAVG, seguido de diagnósticos básicos, para lo cual:

- Realiza el ajuste del modelo lineal
- Evalúa si al menos un predictor es significativo
- Compara varianzas entre grupos
- Realiza el diagnóstico de Gráficos
- Muestra la variación de los residuales
- Verifica si la mediana está cerca de cero

6.1.2. Modelo 2 Modelo de regresión lineal múltiple con selección de variables

```
78  #MODELO 2 Tomando en cuenta todos los regresores
79  lmesh2  <- lm(log(METRIC)~ ..FAILAVG+I(exp( ..FAILAVG))+TXDURATION+PDF+MXN+M_STEP
        +M_PACKET+INTERFACES, data=mesh_fit)
80  summary(lmesh2)
81  ## Stepwize backward de lmesh2
82  lm2d.step <- step(lmesh2,method = "backward")
```

Figura 25. Modelo de regresión múltiple

Este código ajusta un modelo de regresión lineal múltiple con transformaciones y luego aplica selección de variables para simplificarlo, para lo cual:

- Se realiza la transformación logarítmica para normalizar o linealizar relaciones
- Se elimina predictores no significativos
- Se encuentra el modelo más simple que explique similar varianza.

6.1.3. Modelo 3 Modelo propuesto

```
84  #Modelo 3 Modelo propuesto
85  lmesh3  <- lm(log(METRIC) ~  ..FAILAVG + I(exp( ..FAILAVG)) + TXDURATION + MXN +
        M_STEP, data=mesh_fit)
86  summary(lmesh3)
87  anova(lmesh3)
88  par(mfcol = c(2, 2))
89  plot(lmesh3)
90  boxplot(lmesh3$residuals,main='Boxplot: Residuals',ylab='residual value')
```

Figura 26. Modelo propuesto

Este código ajusta un modelo de regresión lineal con predictores seleccionados y transformaciones, seguido de diagnósticos, para lo cual:

- Se realiza una transformación logarítmica para linealizar relaciones o normalizar residuos.
- Se realiza una transformación exponencial para capturar no linealidades.
- Se compara la varianza explicada por cada predictor secuencialmente.
- Se busca patrones no lineales
- Se evalúa la simetría alrededor de cero

- *Validación de modelos de regresión*

```
##Resumen de los modelos
stargazer(lmesh1,lmesh3, type="text", df=FALSE)
anova(lmesh1,lmesh3)
## multicolinealidad del modelo
print(cbind(car::vif(lmesh3)))
## test homedasticidad
ncvTest(lmesh3)
## Selecci n automarica de variables
print(BIC(lmesh3))
```

Figura 27. Validación de modelos de regresión

El código realiza lo siguiente:

- Genera una tabla comparativa de los modelos.
- Muestra coeficientes, errores estándar, significancia estadística, y métricas de ajuste.
- Evalúa si el modelo complejo es estadísticamente mejor que el simple.
- Calcula el Factor de Inflación de la Varianza (VIF) para cada predictor.
- Evalúa si la varianza de los residuales es constante.

6.2. Análisis de Resultados

En base a la configuración planteada en NS-3 se ha podido obtener una base de datos la cual con el uso del RSTUDIO se obtiene los siguientes resultados

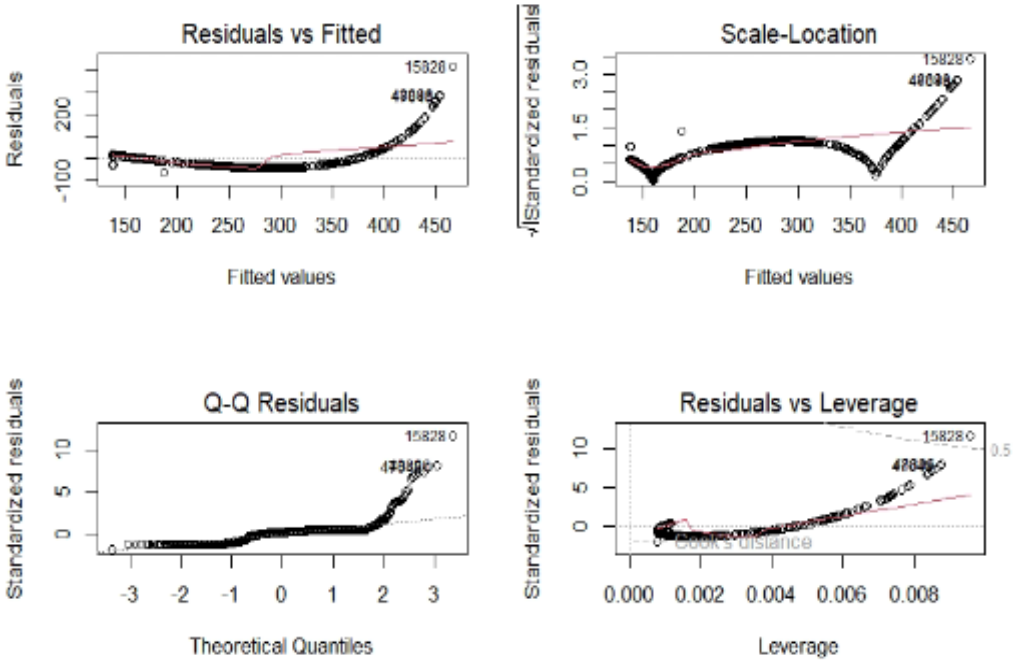

Figura 28. Modelo IEEE 802.11

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.864e+00  2.253e-02  82.729   <2e-16 ***
FAILAVG        -1.231e+00  1.524e-02 -80.770   <2e-16 ***
I(exp(FAILAVG)) 2.039e+00  1.117e-02 182.605   <2e-16 ***
TXDURATION      7.697e-04  1.341e-05  57.381   <2e-16 ***
PDF             6.196e-04  9.027e-04   0.686   0.4926
MXN            -6.496e-05  5.710e-05  -1.138   0.2554
M_STEP         -5.485e-05  3.092e-05  -1.774   0.0763 .
M_PACKET        2.150e-08  1.170e-07   0.184   0.8543
INTERFACES      1.335e-04  4.277e-04   0.312   0.7551
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.011 on 1272 degrees of freedom
Multiple R-squared:  0.9988,    Adjusted R-squared:  0.9988
F-statistic: 1.34e+05 on 8 and 1272 DF,  p-value: < 2.2e-16
```

Tabla 4. Resultados configuración planteada en NS-3

Como se puede observar en la tabla 4 se tiene las variables utilizadas para el modelo matemático y el error está dentro del rango de 0.2

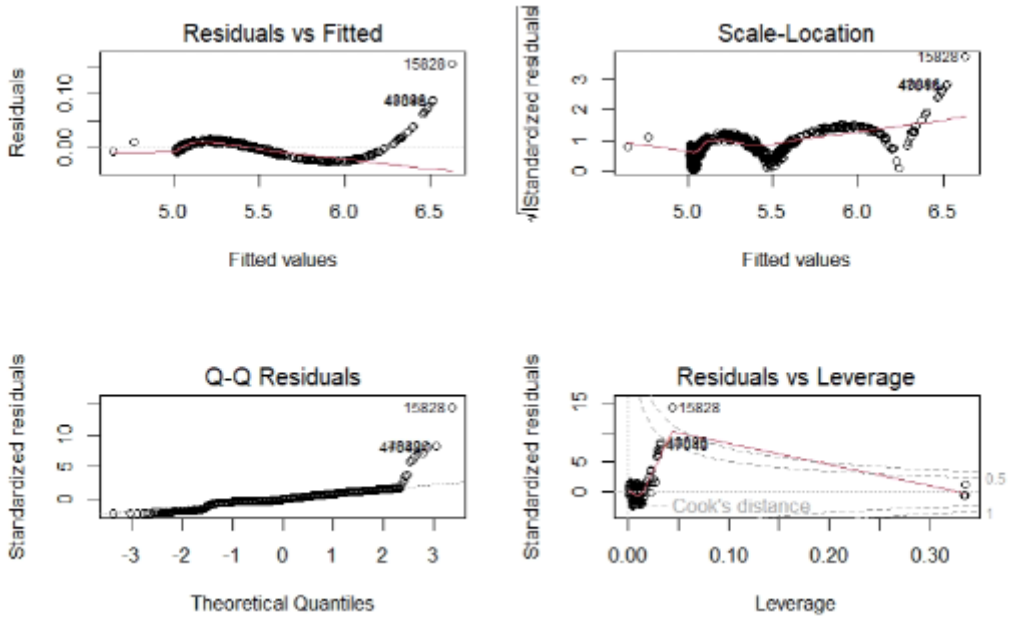

Figura 29. Modelo Propuesto

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.867e+00  2.227e-02  83.824   <2e-16 ***
FAILAVG        -1.230e+00  1.518e-02 -81.001   <2e-16 ***
I(exp(FAILAVG)) 2.038e+00  1.111e-02 183.391   <2e-16 ***
TXDURATION      7.693e-04  1.338e-05  57.504   <2e-16 ***
MXN            -7.856e-05  5.246e-05  -1.498    0.134
M_STEP         -6.013e-05  2.969e-05  -2.025    0.043 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01099 on 1275 degrees of freedom
Multiple R-squared:  0.9988,    Adjusted R-squared:  0.9988
F-statistic: 2.148e+05 on 5 and 1275 DF,  p-value: < 2.2e-16
```

Tabla 5. Resultados configuración modelo propuesto

Como se puede observar en la tabla 5, el modelo propuesto sólo considera 5 variables de las 11 que inicialmente se tenían y en la gráfica se tiene un error menor y el error se encuentra dentro del 0.1.

	Dependent variable:	
	METRIC	log(METRIC)
	(1)	(2)
FAILAVG	398.267***	-1.230***
	(4.896)	(0.015)
I(exp(FAILAVG))		2.038***
		(0.011)
TXDURATION		0.001***
		(0.00001)
MXN		-0.0001
		(0.0001)
M_STEP		-0.0001**
		(0.00003)
Constant	136.523***	1.867***
	(1.229)	(0.022)
Observations	1,281	1,281
R2	0.838	0.999
Adjusted R2	0.838	0.999
Residual Std. Error	35.595	0.011
F Statistic	6,618.366***	214,755.200***
Note:	*p<0.1; **p<0.05; ***p<0.01	

Tabla 6. Resultados comparación entre modelos

En la tabla 6 se puede observar que se realiza una comparación entre los dos modelos donde se puede observar que el modelo 2 presenta mejores resultados para R2, R2 ajustado, Error residual y error estándar residual.

7. Discusión

El análisis de los tres modelos de regresión propuestos permite analizar aspectos importantes del proceso de modelización y la relación entre la simplicidad y la complejidad en los modelos estadísticos.

- **Simplicidad frente a complejidad**

El Modelo 1, al ser el más simple, presenta la ventaja de ser fácil de interpretar y rápido de calcular, sin embargo, existe la probabilidad que no capture toda la complejidad de las relaciones entre las variables, ya que solo considera un predictor. El Modelo 1, es útil como punto de partida, mas, no debe ser considerado una solución definitiva.

El Modelo 2, incluye más variables y transformaciones, por lo que presenta un enfoque más completo que podría proporcionar un ajuste más preciso. Sin embargo, la mayor complejidad puede traer consigo problemas adicionales, los cuales deben ser tratados de manera cuidadosa para equilibrar la precisión y la capacidad de generalización del modelo.

El Modelo 3, al ser una versión simplificada del Modelo 2, intenta mantener el poder explicativo de un modelo más complejo, pero reduciendo la cantidad de variables y transformaciones que pueden ser innecesarias. Al simplificar se reduce el riesgo de sobreajuste.

- **Selección de variables**

La selección de variables es un aspecto importante en la construcción de modelos de regresión. En el Modelo 2, se utilizó un enfoque de selección automática de variables mediante el paso hacia atrás, el cual resulta útil para identificar qué variables realmente aportan valor al modelo.

En el Modelo 3 se reduce la cantidad de regresores, eliminando las variables menos significativas. Se debe considerar que las variables eliminadas pueden tener efectos indirectos o interacciones no detectados con otras variables que podrían influir en la variable dependiente.

- **Interpretabilidad**

El Modelo 2, al incluir tantas variables y transformaciones, se puede volver menos interpretable lo que puede dificultar la presentación de los resultados de manera clara y comprensible.

El Modelo 3, al ser más sencillo, permite una interpretación más directa de las relaciones entre las variables. La simplicidad del Modelo 3 no significa que sea menos efectivo. Se debe considerar que los modelos más simples pueden ofrecer soluciones más claras y comprensibles.

8. Conclusiones

Mediante el uso del programa NS-3 simulamos una red WMN utilizando el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s obteniendo asi las variables de desempeño.

Con el programa RStudio y las variables obtenidas con el programa NS-3 se realizó el análisis y evaluación de los modelos matemáticos propuestos con el objetivo de saber cuál es el modelo matemático que permita mejorar el desempeño de HWMP.

Con lo antes mencionado podemos indicar que de los tres modelos matemáticos el modelo propuesto (modelo 3) utiliza las variables mas relevantes haciéndolo así un modelo mas interpretativo y con menos riesgo de sobreajuste.

Por lo tanto, con el modelo propuesto se ha podido determinar que son 5 métricas las necesarias para mejorar el desempeño del protocolo HWMP.

Métricas	Descripción
FAILAVG	Tasa promedio de error de la trama
TXTDURATION	Tiempo de transmisión de la trama
MXN	Tamaño de la matriz
M_STEP	Distancia entre los nodos
INTERFACE	Numero de interfaces de radio por nodo.

Tabla 7. Métricas utilizadas para el protocolo HWMP

9. Recomendaciones

- Es importante considerar el costo computacional derivado de este trabajo, es decir, cómo afecta la inclusión de nuevas variables y nuevas formas de cálculo. Este problema debe abordarse, de tal manera que se pueda simplificar el cálculo de métricas y aliviar las tareas de procesamiento.

- Es relevante continuar la investigación llevando los modelos propuestos al simulador para una primera evaluación y, si es necesario, continuar con su implementación en escenarios reales.

- Se debe de indicar que en base al estudio realizado se obtuvo las métricas más representativas, pero ante los avances de la tecnología es posible que estas se puedan incrementar o variar.

1. Referencias bibliográficas.

- Acuña Martínez, D., Roncallo Kelsey, R.: Redes inalámbricas enmalladas metropolitanas.pp. 46--91 (2006)

- Gralla, P. (2006), Como funcionan las redes inalámbricas. Madrid, España. Anaya multimedia.

- Roig, O., & Valenzuela, J.; Comes, R. (2003). Principios de comunicaciones móviles. Barcelona, España. UPC.

- Griera et al. (2008). Estructura de redes de computadores. Barcelona, España. UOC.

- Espiga, A. (2012). Selección de portal en redes inalámbricas malladas utilizando aprendizaje estadístico. Recuperado de http://premat.fing.edu.uy/ingenieriamatematica/archivos/tesis_alejandro_espiga.PDF

- Municio, E., Quispe, M., Paucar, R., Chuchón, M., & Díaz, D. (Sin Fecha). Evaluación de protocolos de encaminamiento en una red inalámbrica mallada desplegada en una zona rural. Recuperado de http://oa.upm.es/19959/1/Articulo_Redes_Malladas_URSI2013_19_03_ 03.pdf

- ns-3 project. (2010-2017). Introduction-tutorial. Estados Unidos. Recuperado de https://www.nsnam.org /docs/tutorial/html/introduction.html

- Novoa, A, & Reyes, R. (2007). Análisis, estudio y site survey para investigar la factibilidad con respecto a la cobertura de señal wireless basada en el estándar 802.11 (wi-fi) en el campus sur de la Universidad Politécnica Salesiana. (Título de grado). Universidad Politécnica Salesiana, Quito, Ecuador

- Escudero, A. (2007). Estándares en Tecnologías Inalámbricas. Recuperado de: http://it46.se/courses/wireless/materials/es/02_Estandares/02_es_estandaresinalambricos_guia_v01.pdf.

- Blengio, R; Sosa, G, & Sosa, D (2008). Monitoreo Redes Mesh. Recuperado de: https://iie.fing.edu.uy/publicaciones/2008/SSB08/SSB08.pdf

2. Anexos

CERTIFICADO DE TRADUCCIÓN

Yo, **Nathali del Cisne Cuenca Collaguazo,** con cédula de Identidad **1105775330**, como *Licenciada en Ciencias de la Educación Mención Idioma Inglés*, certifico que este documento de resumen del **Proyecto de Titulación:** ***Evaluación de un modelo matemático para la elección de métricas en el protocolo de ruteo HWMP (Hybrid Wireless Mesh Protocol) perteneciente al estándar IEEE 802.11s, utilizando NS-3 y RStudio***, de autoría de la **Srta. Marilin Salome Quishpi Flores** con **C.I. 0301549143,** *de la Maestría en Telecomunicaciones de la Universidad Nacional de Loja, modalidad en línea,* es una versión correcta de traducción literal del español al inglés. También, se certifica la fidelidad de la traducción más no se asume responsabilidad por la autenticidad o el contenido del documento en la lengua de origen.

Es todo cuanto puedo certificar en honor a la verdad y autorizo hacer uso del presente certificado de la manera ética en lo que a sus intereses convenga.

Miércoles, 04 de abril del 2025.

NATHALI DEL CISNE CUENCA COLLAGUAZO

Mgs. Nathali Cuenca Collaguazo

N. R. SENESCYT: 1008-2018-1987008 - 7241178977
TELF. 07 211 2044
CEL. 0981207483 - 0991032122
EMAIL: nathali161994@hotmail.com